

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

5

Listing of Claims:

1. (Currently amended) A method for allowing objects in a first programming language to communicate with objects in a second programming language without the use of a broker,
10 comprising:

- a) receiving metadata information from a server running said second programming language on a client running said first programming language;
- 15 b) generating proxies for said first programming language on said client from said metadata information, ~~using a development tool for said first programming language,~~ wherein said proxies are generated by a one-to-one mapping of classes from said second programming language to said first programming language and ~~said proxy is customized for the client's intended purpose;~~ and
- 20 c) implementing said proxies on said client, wherein said method is provided solely in said first programming language and said client does not
- 25 require any components from said second programming language.

30

2. (Original) The method according to claim 1, including an additional step d) using said proxies to enable bi-directional communication between said client and said server.

5

3. (Previously Presented) The method according to claim 1, wherein said first programming language is a JAVA cross platform programming language and said second programming language is common language runtime (CLR).

10

4. (Previously Presented) The method according to claim 1, wherein said first programming language is a .Net Remoting programming language and said second programming language is a JAVA cross platform programming language.

15

5. (Original) The method according to claim 1, wherein said client and said server communicate using SOAP formatted messages.

20 6. (Original) The method according to claim 1, wherein said client and said server communicate using binary formatted messages.

7. (Original) The method according to claim 1, including
25 the additional step of passing said proxies to a runtime tool using said first programming language.

8. (Currently Amended) The method according to claim 7,
wherein ~~said runtime tool is capable of operating~~
30 ~~independently of said development tool~~ proxies are

generated using a development tool for said first
programming language .

9. (Currently amended) A brokerless system enabling bi-
5 directional communication using .Net Remoting protocol
between JAVA objects in a JAVA virtual machine environment
and .Net assemblies objects in a common language runtime
(CLR) environment, comprising:

(a) a computer network;

10 (b) a JVM computer having random access memory (RAM)
and at least one of hard disk storage memory
(HDS) and solid state storage memory (SSSM), said
computer having a JAVA Virtual Machine (JVM)
environment and JAVA objects in one of said HDS
15 and SSSM, said JVM computer coupled to said
computer network;

(c) a CLR computer having random access memory (RAM)
and at least one of hard disk storage memory
(HDS) and solid state storage memory (SSSM), said
20 computer having a CLR environment and .Net
assemblies in one of said HDS and SSSM, said CLR
computer coupled to said network;

(d) a JAVA development computer with a RAM, and at
least one of HDS and SSSM, said JAVA development
25 computer having a JVM environment and a JAVA-
based tool in one of said HDS or SSSM, said JAVA
development computer coupled to said network,
~~wherein said JAVA based tool is used, during~~
~~development, to select .Net assemblies running on~~
30 ~~CLR computers), wherein~~

(i) ~~said JAVA-based tool being used during development to select .Net assemblies running on said CLR computer on said computer network and to generate~~ a
5 ~~corresponding set of~~ JAVA proxies corresponding to specified .Net assemblies running on said CLR computer, wherein said JAVA proxies are generated by a one-to-one mapping of classes between JAVA and CLR
10 ~~customized according to their intended purpose; and~~

(ii) said JAVA proxies are copied onto said JVM computer and are operative to allow said
15 specified .Net assemblies on said CLR
computer;

(e) a CLR development computer having memory comprising RAM, and at least one of HDS and SSSM, and having a CLR environment in said memory, said
20 CLR development computer coupled to said computer network, and having a CLR-based tool in said memory ~~operative during development to select specified JAVA objects on said JVM computer over said computer network and to~~ wherein said CLR-
25 based tool generates a corresponding set of CLR .Net proxies corresponding to specified JAVA objects, wherein said .Net proxies are generated by a one-to-one mapping of classes between JAVA and CLR, and wherein said CLR .Net proxies are
30 copied onto said CLR computer and are operative to allow said CLR objects to communicate with

said specified JAVA objects on said JVM computer,
~~said CLR proxies being customized according to~~
~~their intended purpose.~~

5 10. (Currently amended) The system of claim 9, further
comprising a JAVA cross platform programming language based
runtime tool stored on said JVM ~~one~~ computer for handling
said JAVA proxies and said .Net proxies.

10 11. (Previously Presented) The system of claim 10,
wherein said JAVA cross platform programming language based
runtime tool is capable of operating independently of said
JAVA cross platform programming language based tools for
generating JAVA and .Net proxies.

15 12. (New) A computer readable medium having instructions
in a first programming language which, when executed by a
client, enable objects in said first programming language
to communicate with objects in a second programming
20 language without the use of a broker, said instructions
comprising:

- a) receiving metadata information from a server running
said second programming language;
- b) generating proxies on said client from said metadata
25 information, wherein said proxies are generated by a
one-to-one mapping of classes from said second
programming language to said first programming
language; and

30

c) implementing said proxies on said client,
wherein said instructions are provided solely in said first
programming language and said client does not require any
components from said second programming language.

5

13. (New) The computer readable medium according to claim
12, wherein said instructions include an additional step d)
using said proxies to enable bi-directional communication
between said client and said server.

10

14. (New) The computer readable medium according to claim
12, wherein said first programming language is a JAVA cross
platform programming language and said second programming
language is common language runtime (CLR).

15

15. (New) The computer readable medium according to claim
12, wherein said client and said server communicate using
SOAP formatted messages.

20

16. (New) The computer readable medium according to claim
12, wherein said client and said server communicate using
binary formatted messages.

25

17. (New) The computer readable medium according to claim
12, including the additional step of passing said proxies
to a runtime tool using said first programming language.

30

18. (New) The computer readable medium according to claim
17, wherein said proxies are generated using a development
tool for said first programming language.